**Getting started with Image Processing Using OpenCV**

[**Anuradha Mahato**](https://www.analyticsvidhya.com/blog/author/anuradha8220802/)**— Published On March 14, 2023 and Last Modified On March 14th, 2023**

[Advanced](https://www.analyticsvidhya.com/blog/category/advanced/) [Classification](https://www.analyticsvidhya.com/blog/category/classification/) [Computer Vision](https://www.analyticsvidhya.com/blog/category/computer-vision/) [Data Science](https://www.analyticsvidhya.com/blog/category/data-science/) [Image Analysis](https://www.analyticsvidhya.com/blog/category/image-analysis/) [OpenCV](https://www.analyticsvidhya.com/blog/category/opencv/)

Introduction

Image processing is a branch of [computer vision](https://www.analyticsvidhya.com/blog/2021/06/everything-happening-in-computer-vision-that-you-should-know/) that uses various algorithms to manipulate and analyze digital images. It involves the use of mathematical or statistical operations to modify images for many applications, including and not limited to medical and satellite imagery and digital photography. This article explores the fundamentals of image processing and some of the techniques used in this field.

This article was published as a part of the [Data Science Blogathon.](https://datahack.analyticsvidhya.com/blogathon/)

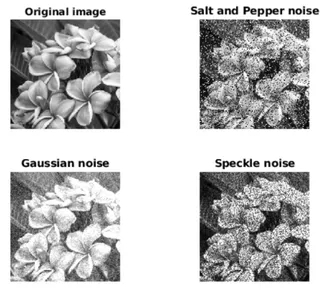
Table of Contents

1. [Fundamentals of Image Processing](https://www.analyticsvidhya.com/blog/2023/03/getting-started-with-image-processing-using-opencv/#Fundamentals_of_Image_Processing)
2. [Applications of Image Processing](https://www.analyticsvidhya.com/blog/2023/03/getting-started-with-image-processing-using-opencv/#Applications_of_Image_Processing)
3. [Techniques for Image Preprocessing](https://www.analyticsvidhya.com/blog/2023/03/getting-started-with-image-processing-using-opencv/#Techniques_for_Image_Preprocessing)
4. [Applying Some Techniques](https://www.analyticsvidhya.com/blog/2023/03/getting-started-with-image-processing-using-opencv/#Applying_Some_Techniques)
5. [Conclusion](https://www.analyticsvidhya.com/blog/2023/03/getting-started-with-image-processing-using-opencv/#Conclusion)

Fundamentals of Image Processing

[Digital images](https://www.analyticsvidhya.com/blog/2021/08/image-processing-in-python-the-computer-vision-techniques/) are broadly made up of pixels, which are tiny boxes representing the color and brightness values at that point in the image. Image processing involves handling these pixels in a desired manner to achieve what is required for the image. Most of the common operations performed on a digital image include filtering, enhancement, restoration, etc.

Filtering is a process of eliminating unwanted noise from an image. It is done by applying a filter that adjusts the image’s pixel values. Based on the type of filter, they can be used for a wide range of applications. They can be designed to remove specific types of noise, such as Gaussian noise, salt-and-pepper noise, or speckle noise. The filters that help in removing the above-mentioned noises include the median filter, the mean filter, and the Gaussian filter.



Enhancement is one process that can improve the quality of an image. It is done by modifying the brightness or contrast of the image. These techniques may be simple, like adjusting the brightness and contrast using a histogram, or more complex, like using algorithms to enhance the edges and textures in an image.



Source: Mathworks.com

[Enroll Now](https://blackbelt.analyticsvidhya.com/plus?utm_source=blog_india&utm_medium=desktop_banner_between_articles&utm_campaign=15-Aug-2023||&utm_content=tools)

Restoration is the process of recovering an image that some noise or other artifacts may degrade. The techniques involve using mathematical methods to estimate the original image from the corrupted version. It is done using techniques such as deconvolution, which is used to get the original image from a blurred version, or denoising, which is used to remove noise from an image.



Source: Mathworks.com

Image preprocessing is quite useful to improve the quality of images and thus boost them for analysis and further processing. Some powerful image preprocessing techniques include noise reduction, contrast enhancement, image resizing, color correction, segmentation, feature extraction, etc. It is an essential step in image analysis that helps enhance the data in images and reduce clutter. As technology continues to advance, image processing will likely become even more important in our daily lives.

Applications of Image Processing

Image preprocessing is a vital step when working with image data. The best results can be obtained when preprocessing of images is done according to the application involved. It is used in various domains, as listed below:

* Medical Imaging to improve the quality of medical images, making it easier to detect diseases or abnormalities
* Object Recognition in images, such as recognizing faces or license plates in surveillance videos
* Object Detection, i.e., primarily used in self-driving cars to navigate the roads better and avoid accidents
* Satellite imagery uses the same for enhancing the image quality for weather forecasting, mapping, etc

Techniques for Image Preprocessing

The choice of techniques depends on the nature of the image and the application. Here are a few techniques to improve image quality and suitability:

* **Noise Reduction**: Noise in an image can be caused by various factors such as low light, sensor noise, and compression artifacts. Noise reduction techniques aim to remove noise from the image while preserving its essential features. Some common noise reduction techniques include Gaussian smoothing, median filtering, and wavelet denoising.
* **Contrast Enhancement**: Contrast enhancement techniques aim to increase the contrast of an image, making it easier to distinguish between different image features. These techniques can be helpful in applications such as medical imaging and surveillance. Some standard contrast enhancement techniques include histogram equalization, adaptive histogram equalization, and contrast stretching.
* **Image Resizing**: Image resizing techniques are used to adjust the size of an image. Resizing can be done to make an image smaller or larger or to change its aspect ratio. Some typical image resizing techniques include nearest neighbor interpolation, bilinear interpolation, and bicubic interpolation.
* **Color Correction**: Color correction techniques are used to adjust the color balance of an image. Color correction is important in applications such as photography, where the color accuracy of an image is critical. Some common color correction techniques include gray world assumption, white balance, and color transfer.
* **Segmentation**: Segmentation techniques are used to divide an image into regions based on its content. Segmentation can be helpful in applications such as medical imaging, where specific structures or organs must be isolated from the image. Some standard segmentation techniques include thresholding, edge detection, and region growing.
* **Feature Extraction**: Feature extraction techniques are used to identify and extract relevant features from an image. These features can be used in object recognition and image classification applications. Some standard feature extraction techniques include edge detection, corner detection, and texture analysis.

Applying Some Techniques

Here are a few image processing techniques that involve grayscaling, thresholding, noise reduction with median and gaussian filters, histogram visualization before and after thresholding, and canny edge detection applied on a sample image.

# Sample downloaded image

**import** cv2

**import** matplotlib.pyplot **as** plt

pic1 = plt.imread('download.jpg')

plt.imshow(pic1)

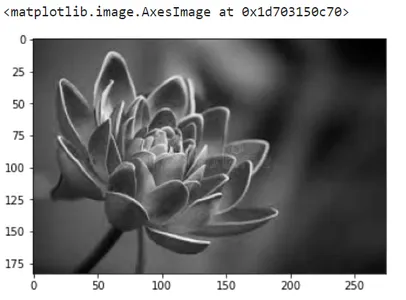


Source: Dreamstime

#Converting the sample image to grayscale

img = cv2.cvtColor(pic1, cv2.COLOR\_BGR2GRAY)

plt.imshow(img,cmap='gray')

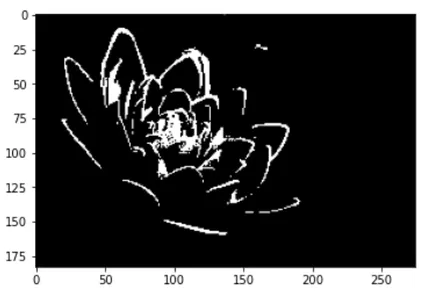


**Thresholding**: Binary threshold output contains only two colors, black and white. It maps all values greater than the threshold to white and less than that to black.

#Thresholding: try playing with the threshold value (144 here) to see the changes

ret, thresh1 = cv2.threshold(img, 140, 255, cv2.THRESH\_BINARY)

plt.imshow(thresh1,cmap='gray')



**Noise Reduction**: It is generally done with filtering according to the nature of the noise. Here, as we don’t know about the nature of noise that may be present, we try applying median and gaussian filters.

#Median filter

img = cv2.cvtColor(pic1, cv2.COLOR\_BGR2GRAY)

median = cv2.medianBlur(img,5)

plt.figure(figsize=(16, 16))

plt.subplot(121),plt.imshow(img,cmap = 'gray')

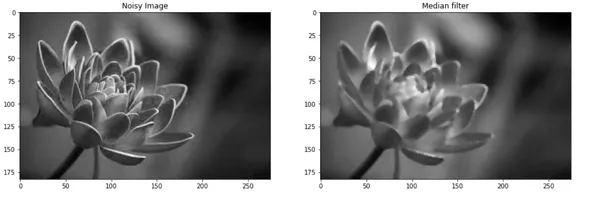
plt.title('Noisy Image')

plt.subplot(122),

plt.imshow(median,cmap = 'gray')

plt.title('Median filter')

plt.show()



gaussian\_blur1 = cv2.GaussianBlur(img,(5,5),2,cv2.BORDER\_DEFAULT)

gaussian\_blur2 = cv2.GaussianBlur(img,(5,5),7,cv2.BORDER\_DEFAULT)

plt.figure(figsize=(20, 20))

plt.subplot(1,3,1),plt.imshow(img,cmap = 'gray')

plt.title('Noisy Image')

plt.subplot(1,3,2),

plt.imshow(gaussian\_blur1,cmap = 'gray')

plt.title('smoothing with Gaussian sigma=2')

plt.subplot(1,3,3),

plt.imshow(gaussian\_blur2,cmap = 'gray')

plt.title('smoothing with Gaussian sigma=7')



**Otsu’s Thresholding**: Here, we don’t specify the threshold value for mapping values to black and white. It uses a histogram to estimate what threshold would work best for the given image and thus is more useful.

#Otsu's thresholding before and after Gaussian filtering

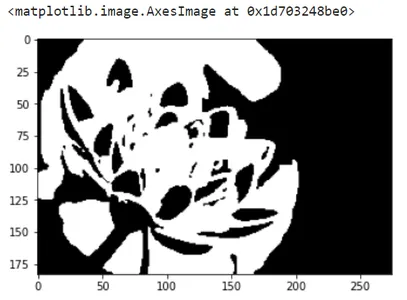
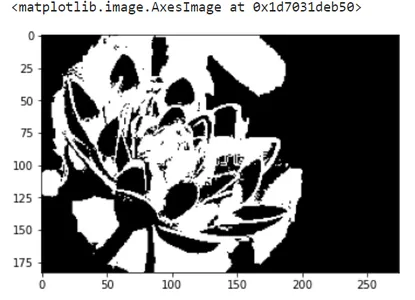
ret2,th2 = cv2.threshold(img,0,255,cv2.THRESH\_BINARY+cv2.THRESH\_OTSU)

blur = cv2.GaussianBlur(img,(5,5),0)

ret3,th3 = cv2.threshold(blur,0,255,cv2.THRESH\_BINARY+cv2.THRESH\_OTSU)

plt.imshow(th2,cmap='gray')

plt.imshow(th3,cmap='gray')



Histogram is a visual representation of the number of pixels of each image’s intensity value. The changes in histograms before and after applying thresholding on original and filtered images are shown below.

plt.figure(figsize=(16,16))

ret1,th1 = cv2.threshold(img,127,255,cv2.THRESH\_BINARY)

# Otsu's thresholding

ret2,th2 = cv2.threshold(img,0,255,cv2.THRESH\_BINARY+cv2.THRESH\_OTSU)

# Otsu's thresholding after Gaussian filtering

blur = cv2.GaussianBlur(img,(5,5),0)

ret3,th3 = cv2.threshold(blur,0,255,cv2.THRESH\_BINARY+cv2.THRESH\_OTSU)

# plot all the images and their histograms

images = [img, 0, th1,

img, 0, th2,

blur, 0, th3]

titles = ['Original Noisy Image','Histogram','Global Thresholding (v=127)',

'Original Noisy Image','Histogram',"Otsu's Thresholding",

'Gaussian filtered Image','Histogram',"Otsu's Thresholding"]

**for** i **in** range(3):

plt.subplot(3,3,i\*3+1),plt.imshow(images[i\*3],'gray')

plt.title(titles[i\*3]), plt.xticks([]), plt.yticks([])

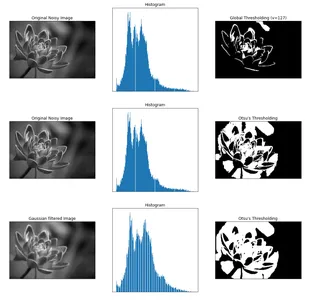
plt.subplot(3,3,i\*3+2),plt.hist(images[i\*3].ravel(),256)

plt.title(titles[i\*3+1]), plt.xticks([]), plt.yticks([])

plt.subplot(3,3,i\*3+3),plt.imshow(images[i\*3+2],'gray')

plt.title(titles[i\*3+2]), plt.xticks([]), plt.yticks([])

plt.show()



**Canny Edge Detection**: It is basically used for edge detection and is built on the Sobel filter. It essentially works upon calculating the image intensity gradient at each pixel of the image, as the gradient is maximum when the color changes quickly in the case of edges.

#Hough Line Transform

dst = cv2.Canny(img, 50, 200, None, 3)

lines = cv2.HoughLines(dst, 1, np.pi / 180, 150, None, 0, 0)

# Draw the lines

**if** lines is **not** None:

**for** i **in** range(0, len(lines)):

rho = lines[i][0][0]

theta = lines[i][0][1]

a = math.cos(theta)

b = math.sin(theta)

x0 = a \* rho

y0 = b \* rho

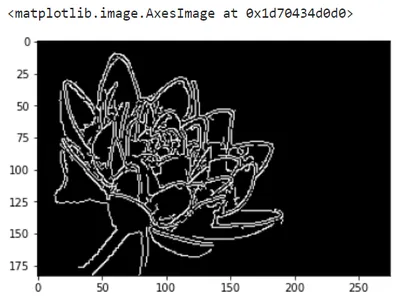
pt1 = (int(x0 + 1000\*(-b)), int(y0 + 1000\*(a)))

pt2 = (int(x0 - 1000\*(-b)), int(y0 - 1000\*(a)))

cv2.line(cdst, pt1, pt2, (0,0,255), 3, cv2.LINE\_AA)

cdst = cv2.cvtColor(dst, cv2.COLOR\_GRAY2BGR)

plt.imshow(cdst)



Conclusion

This article guides you toward the first few steps of image processing. It summarizes some applications that are used in image processing. It is intended to make you familiar with some techniques used in the field and their applications. A few takeaways from the article include:

* Image processing is an essential step in upgrading the quality of the image.
* The wide spectrum of applications includes medical, satellite, object detection, and recognition.
* Filters can help remove noise from the image
* The gradient of an image helps detect the edges in the image

Explore [https://docs.opencv.org](https://docs.opencv.org/) for more beautiful techniques to make your life easier for image processing.

**The media shown in this article is not owned by Analytics Vidhya and is used at the Author’s discretion.**

**Related**

[](https://www.analyticsvidhya.com/blog/2023/02/lets-start-with-image-preprocessing-using-skimage/?relatedposts_hit=1&relatedposts_origin=106036&relatedposts_position=0)

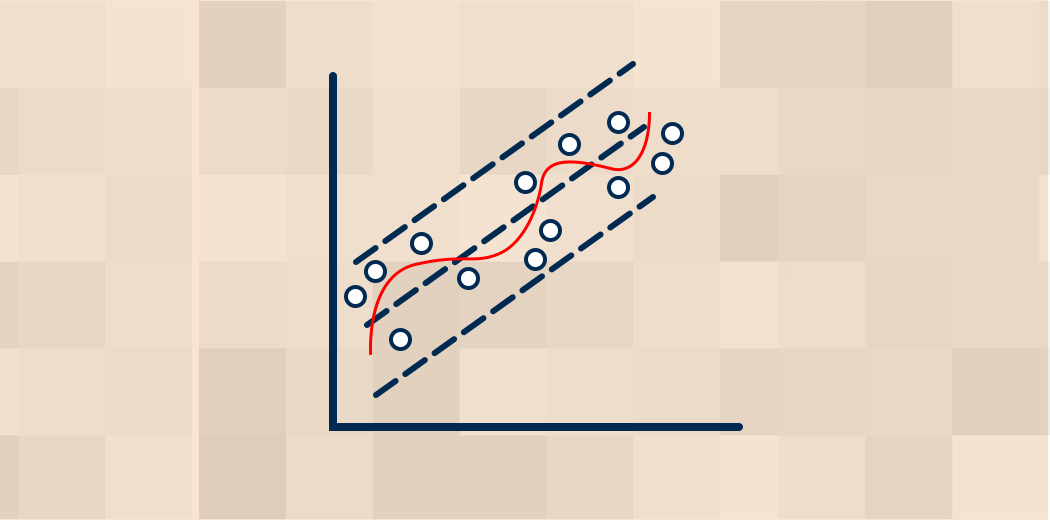
[Let’s Start with Image Preprocessing using SKimage](https://www.analyticsvidhya.com/blog/2023/02/lets-start-with-image-preprocessing-using-skimage/?relatedposts_hit=1&relatedposts_origin=106036&relatedposts_position=0)

[](https://www.analyticsvidhya.com/blog/2021/06/how-to-add-textual-watermarks-to-the-images-with-opencv-and-pil/?relatedposts_hit=1&relatedposts_origin=106036&relatedposts_position=1)

[How To Add Textual Watermarks To The Images With OpenCv and PIL!](https://www.analyticsvidhya.com/blog/2021/06/how-to-add-textual-watermarks-to-the-images-with-opencv-and-pil/?relatedposts_hit=1&relatedposts_origin=106036&relatedposts_position=1)

[](https://www.analyticsvidhya.com/blog/2021/08/image-processing-in-python-the-computer-vision-techniques/?relatedposts_hit=1&relatedposts_origin=106036&relatedposts_position=2)

[Image Processing in Python -The Computer Vision Techniques](https://www.analyticsvidhya.com/blog/2021/08/image-processing-in-python-the-computer-vision-techniques/?relatedposts_hit=1&relatedposts_origin=106036&relatedposts_position=2)

[[](https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/)](https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/)

[Everything you need to Know about Linear Regression!](https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/)

[KAVITA MALI -](https://www.analyticsvidhya.com/blog/author/kavita65/)OCT 04, 2021

[](https://www.analyticsvidhya.com/)

Download App

* Analytics Vidhya

[About Us](https://www.analyticsvidhya.com/about/)[Our Team](https://www.analyticsvidhya.com/team/)[Careers](https://www.analyticsvidhya.com/careers/)[Contact us](https://www.analyticsvidhya.com/contact/)

* Data Scientists

[Blog](https://www.analyticsvidhya.com/blog/)[Hackathon](https://datahack.analyticsvidhya.com/)[Join the Community](https://community.analyticsvidhya.com/?utm_source=Blogs&utm_medium=Footer)[Apply Jobs](https://jobsnew.analyticsvidhya.com/)

* Companies

[Post Jobs](https://www.analyticsvidhya.com/corporate/)[Trainings](https://courses.analyticsvidhya.com/)[Hiring Hackathons](https://datahack.analyticsvidhya.com/)[Advertising](https://www.analyticsvidhya.com/contact/)

* Visit us

© Copyright 2013-2023 Analytics Vidhya.

[Privacy Policy](https://www.analyticsvidhya.com/privacy-policy/) [Terms of Use](https://www.analyticsvidhya.com/terms/) [Refund Policy](https://www.analyticsvidhya.com/refund-policy/)